Eclipse Vorto
Advanced Device Integration
Dr. Olaf Weinmann, Alexander Edelmann
Four scenarios – to be addressed by the Vorto approach

1. Consumers
   ...want to use a large variety of devices in their ecosystem and don't want to be limited to using devices of one specific vendor.
   Flexibility, ease of use

2. Device Vendors
   ...want to increase the number of ecosystems where their devices can be integrated.
   Increase sales

3. Platform Vendors
   ...want to integrate as much as devices as possible into their ecosystem without major efforts.
   Increase sales, become standard

4. Application Developers
   ...want to support a broad range of devices without a need to develop vendor specific code.
   Increase sales, reduction of complexity
Scenario 3 / 4: Platform vendors / application developers

- Bosch M2M
- ThingWorx
- Qivicon
- Device 1
- Device 2
- Device 3
- Device 4
- Device 5
**Eclipse Vorto**

**Information models** – Abstract representations of functionalities, properties, and status of real objects

- **SAMSUNG UE55**
- **Bosch VSC-1516**

**Display**
- Info model: Samsung UE55

**Camera**
- Info model: Bosch VSC-1516
Information models – Abstract representations of functionalities, properties, and status of real objects

Components
- Tuner
- Display
- Speakers
- Motion Detector

Information model
- Function block: Tuner
- Function block: Display
- Function block: Speaker
- Function block: Motion Detector
- Device specific information
Our idea – Creation of a flexible meta model and code generators for generating specific representations
Eclipse Vorto

The project – four components

- **Create representations for different ecosystems**
- **Eclipse Vorto**
- **IoT Tool Set**
- **Information Meta Model**
- **Code Generators**
- **Repository**
- **Create information models**
- **Manage and provide information models**

Create representations for different ecosystems

Standardize the structure of information models
Meta information model – Definition of the structure of information models

Meta Information Model

Eclipse EMF based Meta Information Model

Specifies the structure of derived information models

Info model

Display

Instance of

Camera

Info model

Instance of

Display

Camera
Eclipse Vorto

The IoT Tool Set – Creation of meta model conform information models

Eclipse Platform based IoT Tool Set

Created using

Info model
Display

Created using

Info model
Camera

Created using

Info model
Camera
Eclipse Vorto

The IoT Tool Set – realized as an Eclipse plugin and provides a textual DSL editor

- The grammar of the DSL corresponds to the meta information model
- Auto completion and syntax highlighting allow for comfortably creating information models
- A graphical environment for creating information models shall be added
- The Tool Set shall allow for connecting to the information model repository

Part of the initial contribution
In scope of the project
The central **Information Model Repository** – manage and provide existing information models

When version 1.0 of the repository is released it shall be **hosted by Eclipse** to promote a worldwide standardization of information models.
**Eclipse Vorto**

**Code Generators** – create information model based implementations

The code generator extension point of the IoT Tool Set allows developers to create additional (domain specific) code generators.
The Vorto example code generators

- **eclipse smarthome**: Binding XMLs
- **kura**: Java, XML
- **BOSCH**: Invented for life, Java, XML, XSD
- **jetty://**: Web UI Generator, Java, XML, JS

**IoT Tool Set**

- **Smart Home**
- **Kura**
- **Bosch M2M**
- **Web UI**

**Code Generator Extension Point**

**Information Model Repository**

**Vorto**
Vorto at a glance

Device Manufacturer

Vorto

IoT Tool Set

Meta Information Model

Information Model Repository

Information Model

Platform Vendor

Platform specific Code Generator

Solution Developer

Device

integrates

is described by
creates

browses

_instance_of

reads

writes

invokes

uses

provides

Java

C++

...